CLAIMS

What is claimed is:

1. A method of allocating a processing resources to functions in a queue waiting to be executed, comprising the steps of:

determining an amount of the processor resource available to be assigned;

determining an estimate of an amount of the resource needed for each function waiting in the queue to execute; and

allocating the available resource to the functions based on a hierarchical priority scheme.

2. The method of claim 1, wherein:

the functions are decomposed elements of a more complex process and do not require the same amount of resource to execute.

3. The method of claim 2, wherein:

multiple instances of any function within the process may be invoked by the processor to execute concurrently.

4. The method of claim 3, wherein:

each of the functions within the process is assigned a separate priority within the hierarchical priority scheme.

5. The method of claim 4, wherein:

each instance of each function within the process is assigned a separate priority within the hierarchical priority scheme.

6. The method of claim 2, further comprising the steps of:

assigning a resource throttling value to each function waiting in the queue to be executed, wherein the throttling value determines the reduction of the resource allocated to each of the functions.

7. The method of claim 1, wherein:

the allocation of the available resource to the functions waiting in the queue is conducted to optimize the amount of the resource assigned to these functions.

8. The method of claim 1, wherein:

the allocation of the available resource to the functions waiting in the queue is conducted to optimize a combined number of instances of each function concurrently executed.

9. The method of claim 1, further comprising the steps of:

measuring the actual amount of the resource used;

revising the estimate of the amount of the resource needed for each function waiting in the queue to execute based on the measured amount of the resource used; and

reallocating the available amount of the resource to the functions in accordance with the revised estimate and the hierarchical priority scheme.

10. The method of claim 9, further comprising the steps of:

comparing the measured amount of the resource used to a high and a low threshold value;

setting an alarm if the measured amount of the resource used exceeds the high threshold value; and

removing the alarm if the measured amount of the resource used is less than the

low threshold value.

11. The method of claim 10, further comprising the step of:

assigning a resource throttling value to each function waiting in the queue to be executed when the alarm is set, wherein the throttling value determines the reduction of the resource allocated to each function.

12. The method of claim 10, further comprising the step of:

reducing the number of instances in which a particular function may execute concurrently when the alarm is set.

13. The method of claim 5, further comprising the steps of:

measuring the actual amount of the resource used;

revising the estimate of the amount of the resource needed for each function waiting in the queue to execute based on the measured amount of the resource used; and

reallocating the available amount of the resource to the instances in accordance with the revised estimate and the hierarchical priority scheme.

14. The method of claim 13, further comprising the steps of:

comparing the measured amount of the resource used to a high and a low threshold value;

setting an alarm if the measured amount of the resource used exceeds the high threshold value; and

removing the alarm if the measured amount of the resource used is less than the low threshold value.

15. The method of claim 14, further comprising the step of:

assigning a resource throttling value to each instance of each function waiting in the queue to be executed when the alarm is set, wherein the throttling value determines the reduction of the resource allocated to each instance of each of the functions.

16. The method of claim 14, further comprising the step of:

reducing the number of instances in which a particular function may execute concurrently when the alarm is set.

17. A method of allocating a processing resources to functions in a queue waiting to be executed, comprising the steps of:

determining an amount of the processor resource available to be assigned;

for each of j instances of k functions, calculating a product obtained by:

(a) estimating the amount of resource needed to support the execution of the j^{th} instance of the k^{th} function;

(b) assigning a value of either zero or one to a multiplicand associated with the j^{th} instance of the k^{th} function; and

(c) multiplying the estimated amount of resource needed to support the execution of the j^{th} instance of the k^{th} function by its associated multiplicand and assigning the result to the product associated with the j^{th} instance of the k^{th} function;

for each of the j instances, calculating a sub-total sum obtained by:

(d) summing together the products associated with each of the k functions of the j^{th} instance; and

(e) adding an estimate of the resource needed to support background processing associated with the j^{th} instance to the sum of the products associated with each of the k functions of the j^{th} instance and assigning the result to the sub-total for the j^{th} instance; and

allocating the available resource to the k functions of the j instances based on a hierarchical priority scheme.

18. The method of claim 17, wherein:

the multiplicand value associated with the j^{th} instance of the k^{th} function is determined according to the hierarchical priority scheme.

19. The method of claim 17, further comprising the step of:

repeating the steps recited in claim 17 for each of a number of sequential time periods.

20. The method of claim 19, wherein:

the length of each time period is variable and is no longer than the period needed to execute any one of the j instances of the k functions that are executing concurrently.

21. The method of claim 19, further comprising the step of:

for each of the j instances of the k^{th} function, assigning increasingly higher priority in accordance with an increasingly greater number of time periods that have passed since the j^{th} instance of the k^{th} function was last executed.

22. The method of claim 17, further comprising the steps of:

measuring the actual amount of the resource used;

revising the estimate of the amount of the resource needed for each function waiting in the queue to execute based on the measured amount of the resource used; and

reallocating the available amount of the resource to the instances of each function in accordance with the revised estimate and the hierarchical priority scheme.

23. The method of claim 22, further comprising the steps of:

comparing the measured amount of the resource used to a high and a low threshold value;

setting an alarm if the measured amount of the resource used exceeds the high threshold value; and

removing the alarm if the measured amount of the resource used is less than the low threshold value.

24. The method of claim 23, further comprising the step of:

assigning a resource throttling value to each instance of each function waiting in the queue to be executed when the alarm is set, wherein the throttling value determines the reduction of the resource allocated to each instance of each of the functions.

25. The method of claim 23, further comprising the step of:

reducing the number of instances in which a particular function may execute concurrently when the alarm is set.